

## 周报

本周在实现相似哈希体数据压缩的最后几个步骤，细节如下。

### 思路一（按照 google 相似哈希相似查询的算法）

逐块获得体数据的相似哈希指纹后

首先进行指纹集合的复制，置换，并对复制和置换后的结果进行排序得到有序的指纹查找集合。假设当前指纹集合为 $Q$ ，指纹长度是 $16*5$ 比特，相似比较阈值 $k$ 为4。首先对 $Q$ 中的所有指纹，逐条地使用5个置换函数进行复制置换，结束复制置换后，得到5个和 $Q$ 同样大小的指纹集合，记为 $Q_1, Q_2, Q_3, Q_4, Q_5$ 。 $Q_1$ 和 $Q$ 一样（可认为对应的置换函数将第一个16比特置换到第一个16比特）， $Q_2$ 则是将第二个16比特置换到第一个，依次类推。在进行查询时，将待查询指纹 $f$ 分别与 $Q_1 \sim Q_5$ 集合中的指纹进行查询，查询前，分别对 $f$ 做对应 $Q_i$ 的置换操作。由抽屉原理可知，比较阈值为4，也就是不同的比特位为4，而所有指纹被划分为5个部分，至少必有一个部分是完全相似的。因而，通过上述比较先对前16个比特进行精确匹配，然后将精确匹配后的（事实上，缩小了待查询指纹集合的范围）指纹提取出来进行下一步的检测盒匹配，这是一种空间换时间的算法。

本方法查询部分已经实现，但没有进一步做结果，原因在于：

## 思路二 类向量量化分类

在实现思路一的过程中，结合之前向量量化体数据压缩的实现过程，我觉得可以跳过 google 的那种做法。原因有二，其一是因为在这一阶段，我们的主要目的是生成码表，而不是进行相似查询，而生成码表的方法完全可以借鉴向量量化的形式；其二，相似哈希相似查询那种算法有一个前提，指纹集合  $Q$  包含海量的指纹数量，比如 80 亿，而具体到体数据压缩，假设体数据原始尺度是  $1024^3$ ，块大小是  $4*4*4$ （最后的调试测验结果，体数据块尺度可能会更高），最后包含的体数据块也不过 16777216 个，亦即最后包含的指纹是 16777216 个，远少于 google 相似哈希中的应用，使得直接借鉴向量量化的分类方法成为可能。

目前这一思路已基本实现，我的程序也使用这一思路进行分类和下一步的码表生成，正在测试中，预计两到三个工作日可分析压缩结果。